

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student:

Marek Duda

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: AUTEL, a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Mgr. Jiří Dvorský, Ph.D.**

Konzultant bakalářské práce: Ing. David Raszka

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Šnášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2016

.....*Dvůr*.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Třinci - Oldřichovicích 29. dubna 2016



AUTEL, a. s.
Oldřichovice 790
739 61 Třinec I

Rád bych na tomto místě poděkoval vedení firmy Autel, a.s. za umožnění vykonávat u nich bakalářskou praxi, zejména Ing. Davidu Raszkovi za poskytnutí odborné pomoci při plnění úkolů na praxi. Dále bych chtěl poděkovat panu doc. Mgr. Jiřímu Dvorskému, Ph.D. za vedení při vytváření bakalářské práce.

Abstrakt

V této bakalářské práci popisuji průběh mé činnosti ve firmě Autel, a.s. a její odbornou činnost v oblasti IT. Dále popisuji úkoly které jsem dostal během mého působení na praxi a jejich řešení pomocí frameworku Microsoft.NET, v jazyce C# v programovacím nástroji Microsoft Visual Studio.

Klíčová slova: Microsoft .Net, C#, Microsoft Visual Studio, SerialPort, Inmat51, Inmat66, RS232C, RS485, ORM, Třívrstvá architektura, Oracle SQL Developer Data Modeler, Microsoft SQL Server Management Studio, LINQ

Abstract

In this thesis I describe the course of my work in the company Autel, a.s. and its expertise in IT. I also describe the tasks that I have received during my time on the practice and its solutions using the framework Microsoft.NET , in C# language in programming tool Microsoft Visual Studio.

Key Words: Microsoft .Net, C#, Microsoft Visual Studio, SerialPort, Inmat51, Inmat66, RS232C, RS485, ORM, Three-tier Architecture, Oracle SQL Developer Data Modeler, Microsoft SQL Server Management Studio, LINQ

Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
Seznam výpisů zdrojového kódu	10
1 Úvod	11
2 Odborné zaměření firmy a pracovní zařazení studenta	12
2.1 Odborné zaměření firmy	12
2.2 Pracovní zařazení studenta	12
3 Seznam úkolů zadaných studentovi v průběhu odborné praxe a vyjádření jejich časové náročnosti	14
3.1 Seznámení se s měřicími zařízeními Inmat 51 a Inmat 66	14
3.2 Komunikace po sériovém kanálu, a její implementace na těchto zařízeních	14
3.3 Vytvoření vlastní ORM pro vkládání a získávání dat z firemní databáze	14
3.4 Vytvoření výsledné služby pro obsluhu a sběr dat na zařízeních Inmat 51 a Inmat 66	14
4 Zvolený postup řešení zadaných úkolů	15
4.1 Seznámení se s měřicími zařízeními Inmat 51 a Inmat 66	15
4.2 Komunikace po sériovém kanálu, a její implementace na těchto zařízeních	16
4.3 Vytvoření vlastní ORM pro vkládání a získávání dat z firemní databáze	21
4.4 Vytvoření výsledné služby pro obsluhu a sběr dat na zařízeních Inmat 51 a Inmat 66	22
5 Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe	24
6 Znalosti či dovednosti scházející v průběhu odborné praxe	25
7 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení	26
Literatura	27
8 Přílohy	28

Seznam použitých zkratk a symbolů

IP 65	– Stupeň krytí
LINQ	– Integrovaný jazyk pro dotazování
MES	– Manufacturing execution system
NN	– Nízké napětí
ORM	– Objektově relační mapování
PROFIBUS	– Průmyslová sběrnice určena pro automatizaci
SŘBD	– Systém řízení báze dat
VN	– Vysoké napětí

Seznam obrázků

1	Inmat 51 [4] a Inmat 66[7]	16
2	PROFIBUS vrstva 2	17
3	Vrstva 7 - čtení bloku hodnot z matice hodnot	19
4	Vrstva 7 - zápis bloku hodnot do matice	20

Seznam výpisů zdrojového kódu

1	Implementace telegramu s proměnnou délkou informačního pole	18
2	Implementace čtení bloku hodnot z matice na datové vrstvě	19
3	Implementace zápisu bloku hodnot do matice - nastavení data a času	20
4	Funkce pro nastavení data a času ve čtyřech bajtech	20

1 Úvod

Tuto bakalářskou práci jsem se rozhodl vykonávat jako odbornou praxi ve firmě, jelikož mi přišlo více přínosné pracovat v odborně zaměřené firmě přes IT technologie a to z toho důvodu, abych mohl nabrat nové zkušenosti s programováním ve firmě, které určitě využiji po následném dokončení studia. Pro vykonávání praxe jsem si vybral firmu Autel, a.s. po doporučení mého kamaráda, který v ní pracuje. Na začátku této bakalářské práce, nejdříve popíšu odborné zaměření firmy, její působení v oblasti průmyslů a čím se zabývá její IT oddělení. V další části bakalářské práce již popisuji moji odbornou činnost ve výše uvedené firmě, úkoly které jsem dostal během působení na praxi a postup jak jsem je řešil.

2 Odborné zaměření firmy a pracovní zařazení studenta

2.1 Odborné zaměření firmy

Autel, a.s. je inženýrsko-dodavatelská firma[1], která sídlí v Třinci Oldřichovicích. Zajišťuje spolehlivé a pružné služby a dodávky nejmodernějších technologií na klíč v různých oblastech průmyslu. Mezi tyto služby patří:

- VN, NN technologický silnoproud a regulované pohony,
- prostředky MaR, tzv. „polní instrumentace“,
- řídicí a vizualizační systémy základní úrovně a
- rozvrhování a řízení výroby (MES).

Firma také nabízí široký rozsah inženýringových produktů:

- konzultace ohledně projektů, vedení projektů, zpracování projektové dokumentace, analýza a programování,
- testování zařízení u zhotovitele, instalace, studené a teplé zkoušky, školení operátorů a
- zkušební provoz, závěrečný test projektů a garanční a po garanční servis.

Společnost má rozsáhlé praktické zkušenosti s průmyslovou automatizací v těchto tržních segmentech:

- Hutní průmysl
- Energetika
- Petrochemie a chemický průmysl
- Ekologické stavby
- Zpracovatelský průmysl surovin
- Celulóza a papír

2.2 Pracovní zařazení studenta

Byl jsem zařazen na IT oddělení firmy, které se zabývá vývojem softwaru na zakázku, postaveném na platformě Microsoft .NET. Na tomto oddělení je široká tvorba různých aplikací jak už od jednoúčelových aplikací, které jsou určené pro jednoho uživatele, až po větší informační systémy. Mezi tyto informační systémy patří zejména:

- MES systémy,

- firemní informační systémy,
- docházkové systémy a
- přístupové systémy.

Po přidělení na toto oddělení a seznámení se sním jsem obdržel úkoly, které spadají do MES systémů, a to je seznámení se s měřicími zařízeními, která jsou určena pro sběr dat na úrovni bajtové komunikace, a naprogramování služeb obsluhy pro zařízení Inmat51 a Inmat66.

3 Seznam úkolů zadaných studentovi v průběhu odborné praxe a vyjádření jejich časové náročnosti

3.1 Seznámení se s měřicími zařízeními Inmat 51 a Inmat 66

Mezi mé úplně první úkoly na praxi bylo seznámení se s funkcí měřících zařízení a to zejména s měřicími zařízeními typu Inmat. Dostal jsem tedy sadu katalogových listů a návodů k výrobkům, abych mohl zjistit, k čemu jsou tato měřicí zařízení a jak probíhá jejich funkce. Tento počáteční úkol mi zabral přibližně pět dnů.

3.2 Komunikace po sériovém kanálu, a její implementace na těchto zařízeních

Po nastudování všech potřebných informací o měřících zařízeních typu Inmat, jsem dostal za úkol si nastudovat jak probíhá komunikace po sériovém kanálu v prostředí Microsoft .NET a naimplementovat ji. Celková implementace této části mi trvala přibližně patnáct dnů.

3.3 Vytvoření vlastní ORM pro vkládání a získávání dat z firemní databáze

Abych mohl komunikovat z databází firmy, musel jsem si vytvořit vlastní ORM, které jsem následně otestoval a doladil jeho funkčnost. Celková délka tvorby a testování mi zabrala zhruba deset dnů.

3.4 Vytvoření výsledné služby pro obsluhu a sběr dat na zařízeních Inmat 51 a Inmat 66

Nakonec bylo mým úkolem vytvořit výslednou službu, která bude obsluhovat měřicí zařízení Inmat 51 a Inmat 66. Tato služba má za úkol provádět sběr dat na úrovni bajtové komunikace, která bude probíhat dle jejich protokolu po sběrnici RS485/232C a zpracovávat rámce jejich odpovědi. Pro vkládání bude použito předem vytvořené vlastní ORM s předchozího úkolu. V službě budou ověřené veškeré možné chybové stavy, tak aby byla zajištěna stabilita služby pro použití v průmyslovém nasazení. Plnění tohoto úkolu mi zabralo přibližně dvacet dnů.

4 Zvolený postup řešení zadaných úkolů

4.1 Seznámení se z měřicími zařízeními Inmat 51 a Inmat 66

Inmat 51 [2] a Inmat 66 [5] jsou matematické členy, neboli pracovní měřidla určena k průmyslovému zpracování různých fyzikálních veličin jako jsou elektrické proudy, odpory, frekvence, teploty, průtoky energie, výšky hladiny, vzdálenosti, rychlosti, otáčky a další. Používají se k měření a sběru dat těchto fyzikálních veličin zejména v energetice, plynárenství, teplárenství, v naftařském průmyslu, strojírenství, zemědělství a v chemickém průmyslu.

Inmat 51 [3] se používá k měření průtoku vody, tepla předaného vodou, průtoku vodní páry, tepla předaného vodní párou a následně jako matematický člen stavového přepočítavače plynů. Inmat 66 [6] je ještě rozšířen oproti předchozímu pracovnímu měřidlu o měření chladu, průtoku technických kapalin a k regulaci hladiny v kotli.

Obě pracovní měřidla jsou umístěna v plastové skříní kompaktní konstrukce, která je krytá dvěma víky. Tato konstrukce má průmyslové krytí IP 65, které zajišťuje odolnost elektrospotřebiče před nebezpečným dotykem jakoukoliv pomůckou, vniknutím prachu a je chráněn proti tryskající vodě. Tato konstrukce se skládá ze dvou částí:

- napájecí zdroj se svorkovnicí a
- hlavní deska s analogovými obvody a mikropočítačem.

Oba měřáky umějí archivovat data až do 2 MB, vytvářet rozsáhlou autodiagnostiku a lze je použít i jako samostatná měřidla k vybudování rozsáhlých měřících systémů. Inmat 66 má navíc možnost víceokruhového měření a měření průtoku různých médií v jednom přístroji současně.

Pro výstupní signály slouží u obou měřidel jeden dvou řádkový, šestnácti znakový LCD display, analogové výstupní signály, relé se spínacím kontaktem pro autodiagnostiku, galvanický oddělený impulzní výstup a galvanický oddělený zdroj.

Jako komunikační rozhraní se používá komunikační protokoly RS485, RS232C/U, RS232C. V mém případě jsem se zajímal zejména o komunikační protokol RS485, který je obousměrný a slouží jak pro vstup, tak i pro výstup. Přes tento protokol, lze zapojit až třicet zařízení bez opakovacího pro komunikaci s počítačem. Lze přes něj nastavovat některé parametry jako datum, čas, uživatelské konstanty a parametry u všech komunikačních protokolů. Pomocí RS485 lze číst naměřené a vyhodnocované údaje včetně údajů, které jsou archivovány.

Princip funkce: Tyto měřicí ústředny řídí šestnáctibitový jednočipový mikropočítač. Vstupní fyzikální veličiny se získávají pomocí standardních rozhraní jako je proudová smyčka (0 - 20 mA, nebo 4 - 20 mA), odporových snímačů pro měření teploty a pomocí galvanicky oddělených frekvenčních a impulzních vstupů. Z těchto naměřených hodnot se vyjadřuje jejich hodnota ve fyzikálních jednotkách a vytvářejí se systémové proměnné. Z těchto systémových proměnných je ústředna schopna vypočítat libovolné fyzikální veličiny, jsou-li specifikovány výpočtovým vzorcem nebo tabulkou. Tyto výpočtové vzorce nebo tabulky, jsou volně programovatelné uživatelem



Obrázek 1: Inmat 51 [4] a Inmat 66[7]

pomocí počítače. Stejným způsobem jsou specifikovány veličiny pro sumarizaci, kalibrační konstanty a nelinearity vstupních čidel, povolené meze vstupních i výstupních veličin, rozsahy výstupních proudových signálů, a formáty zobrazení a doprovodný text pro požadované výstupní veličiny. Sumy jsou ukládané do paměti a zálohovány proti výpadku napájení po dobu minimálně 10 let. Vypočtené veličiny, lze zobrazovat na displeji zařízení. Dále je na těchto zařízeních umožněno nastavovat uživatelské konstanty, a toto nastavení lze provést po dohodě s výrobcem pomocí komunikačního rozhraní RS485. Změna těchto uživatelských konstant se může chránit hesly. Uživatelskými konstantami lze nastavovat tyto parametry:

- rozsahy snímačů,
- rozsah proudového výstupu a
- meze pro signalizaci.

4.2 Komunikace po sériovém kanálu, a její implementace na těchto zařízeních

Po nastudování principu funkce měřících ústředen Inmat 51 a Inmat 66, jsem propojil tato zařízení se svým počítačem. Propojení jsem provedl pomocí sériového kabelu typu U232 - P9, který šel následně do převodníku Moxa TCC-80-DB9, který převáděl signál pomocí komunikačního protokolu RS485, a tento převodník jsem zapojil na měřící ústřednu pomocí pinů, dle manuálu. Po nainstalování ovladačů v mém počítači jsem spustil software, který byl dodán k měřícím zařízením typu Inmat a jednalo se o SWK46603. Tento program zpřístupňoval základní komunikaci a servis na těchto zařízeních. Díky tomuto softwaru jsem si mohl vyhledat, na které adrese se nachází připojena ústředna, tuto adresu jsem později použil pro implementaci komunikace se zařízením po sériovém kanálu. Dále tento servisní software zobrazoval všechny důležité systémové proměnné, jako jsou sumy, maxima, uživatelské proměnné a uživatelské konstanty, které se na zařízení měří či dají nastavit.

Následně jsem tedy přistoupil k implementaci komunikace po sériovém kanálu v jazyce C#, na platformě Microsoft .NET. K tomuto účelu jsem si naimplementoval třídu, která vycházela ze třídy SerialPort s knihovny System.IO.Ports. Do konstruktoru této třídy jsem tedy nastavil nezbytné parametry:

- název comportu na kterém bylo zařízení připojeno k počítači, v mém případě COM3,

- přenosovou rychlost, která je v mém případě 9600 Bd,
- paritní bit, u zařízení typu Inmat je sudá parita,
- počet datových bitů, v tomto případě 8 bitů a
- jeden stop bit.

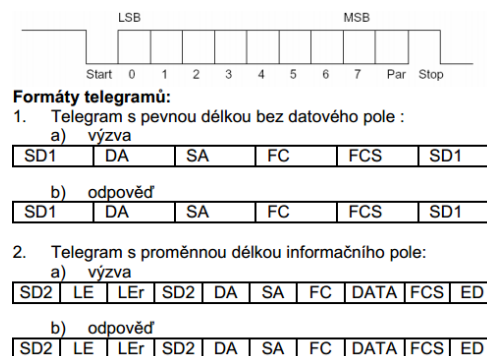
Dále jsem použil metody `Open()` a `Close()` pro otevření a zavření sériového přenosu. Pro příjem dat jsem si naimplementoval událost `DataReceived`.

Dalším krokem bylo naimplementovat samotný komunikační protokol RS485 [8], který je typu master-slave a umožňuje komunikaci mezi zařízeními v obou směrech. Tento komunikační protokol se skládá ze dvou vrstev:

1. PROFIBUS vrstva 2.

Tato vrstva obsahuje dva typy telegramů:

- telegram s pevnou délkou bez datového pole a
- telegram s proměnnou délkou informačního pole.



Obrázek 2: PROFIBUS vrstva 2

Každý telegram se skládá ze znaků, každý znak má délku 11 bitů. Začíná start bitem, následuje 8 datových bitů, sudý paritní bit a stop bit. Vysílá se nejdříve nejméně významný bit. Tyto parametry jsem implementoval do konstruktoru třídy `SerialPort`, jenž jsem popsal výše. Oba typy telegramů fungují tak, že zašlou výzvu na měřicí zařízení a potom přijmou odpověď. Význam použitých symbolů v telegramech:

- SD1 začátek rámce u telegramu s pevnou délkou, zapisuje se kódem 0x10H,
- SD2 začátek rámce u telegramu s proměnnou délkou datového pole, zapisuje se kódem 0x68H,
- LE délka informačního pole, je v ní započítáno DA, SA, FC a DATA, nejnižší hodnota je 4 a nejvyšší je 249,

- LER opakování délky informačního pole,
- DA, SA adresa stanice (SA - zdrojová, DA - cílová), leží v rozmezí 0 - 127 u Inmatu je maximální adresa omezena pouze na 63, jelikož neumí rozšířit adresu pomocí bitu EXT jak je definováno v PROFIBUSu,
- FC řídicí bajt, obsahuje přenosovou funkci a informaci zabraňující ztrátě nebo zdvojení zprávy,
- Data pole dat o délce maximálně 246 Bytů, definováno ve vrstvě 7,
- FCS kontrolní součet, je daný aritmetickým součtem dat informačního rámce, tedy součtem DA, SA, FC a DATA s integrováním přenosu a
- ED konec rámce, hodnota 0x16H.

```
byte[] query = (new byte[] { 0x68, 0x0F, 0x0F, 0x68, (byte)meter.Adr, (
    byte)13, 0x4D }).Concat(mesQuery).Concat(new byte[] { 0xFF, 0x16 }).
ToArray();
```

Výpis 1: Implementace telegramu s proměnnou délkou informačního pole

2. DB-NET vrstva 7.

Jedná se o datovou vrstvu PROFIBUSu, která implementuje protokol DB-NET. Poskytuje tyto služby:

- čtení identifikace přístroje,
- čtení jedné hodnoty,
- čtení jedné hodnoty z matice hodnot,
- čtení bloku hodnot z matice hodnot,
- zápis jedné hodnoty,
- zápis jedné hodnoty do matice hodnot,
- zápis bloku hodnot do matice,
- čtení paměti od zadané adresy a
- zápis do paměti od zadané adresy.

V mém případě jsem použil tyto dvě služby:

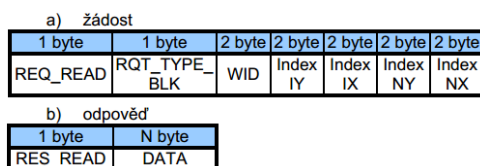
- Čtení bloku hodnot z matice hodnot
Touto službou jsem v datové vrstvě posílal žádost na měřicí zařízení pro získání systémových proměnných, výpočtových proměnných, sum, a pro dotazování jaké je datum a čas na zařízení. Zápis této služby v kódu je následující:

```

int wid = meter.Adr * 1000 + phaseByte;
if (phaseByte != 0x10)
    mesQuery = new byte[] {
        0x01,           //REQ_READ
        0x22,           //RQT_FLOAT_BLK
        (byte)wid,       //WID low
        (byte)(wid>>8), //WID high
        0,0,            //radek
        0,0,            //sloupec
        (byte)(meter.profiles.Where(p => p.Phase == (int)
            phase).Max(p => p.Adr) + 1), 0, //Pocet radku -
            nejvyssi pozadovana adresa + jeden radek
        1,0             //pocet sloupcu
    };

```

Výpis 2: Implementace čtení bloku hodnot z matice na datové vrstvě



Obrázek 3: Vrstva 7 - čtení bloku hodnot z matice hodnot

Význam použitých symbolů:

- REQ_READ žádost na posláni dat, 0x01H,
 - RQT_TYPE_BLK typ žádaných proměnných, int 0x20H, long 0x21H, float 0x22H, string 0x23H,
 - WID identifikátor databázové proměnné v rámci celé sítě, $WID = (DA * 1000D) + INX$,
 - INX index databázové proměnné v rámci stanice, čas a datum 0x10H, systémové proměnné 0x20H, výpočtové proměnné 0x21H, sumy 0x22H,
 - Index IY index řádku databázové proměnné typu matice,
 - Index IX index sloupce databázové proměnné typu matice,
 - Index NY počet řádku databázové proměnné typu matice,
 - Index NX počet sloupců databázové proměnné typu matice a
 - REQ_WRITE žádost na zápis dat, 0x02H.
- Zápis bloku hodnot do matice

a) žádost

1 byte	1 byte	2 byte	2 byte	2 byte	2 byte	2 byte	N byte
REQ_WRITE	RQT_TYPE_BLK	WID	Index IY	Index IX	Index NY	Index NX	DATA

Obrázek 4: Vrstva 7 - zápis bloku hodnot do matice

Tuto službu datové vrstvy jsem používal pro nastavení data a času na měřícím zařízení. Čas a datum je u Inmatu 51 a Inmatu 66 uložen v osmi bajtech. Příkaz pro nastavení data a času vypadá takto:

```

wid = meter.Adr * 1000 + 0x10;
buf = new byte[] { //vyzva - systemova cast
    0x02,          //REQ_WRITE
    0x20,          //RQT_INT_BLK
    (byte)wid,      //WID low
    (byte)(wid>>8), //WID high
    0,0,           //radek
    0,0,           //sloupec
    7,0,           //pocet radku
    1,0,           //pocet sloupcu
    s,0,           //sekundy
    mi,0,          //minuty
    hour,0,        //hodiny
    dayOfWeek,0,   //den tydne
    day,0,         //den mesice
    month,0,       //mesic
    year,0,        //rok

```

Výpis 3: Implementace zápisu bloku hodnot do matice - nastavení data a času

Dále jsem však musel vyřešit na žádost firmy, jak provést toto nastavení času pouze ve čtyřech bajtech a to kvůli případné budoucí úpravě této výsledné služby na zařízeních jako je Inmat 57. U těchto zařízení jde totiž datum a čas nastavit pouze v těchto čtyřech bajtech. Jelikož sekundy, minuty a roky lze nastavit v šesti bitech, hodiny a dny v pěti bitech a měsíce pouze ve čtyřech bitech tak se nám výsledné datum s časem lehce vmístí pouze do čtyřech bajtů. Na tento problém jsem tedy použil bitový posuv.

```

public static byte[] ToPKtimeBytes(this DateTime dtNow)
{
    byte date1 = (byte)((dtNow.Second << 2) | (dtNow.Minute >>
        4));

```

```

byte date2 = (byte)((dtNow.Minute << 4) | (dtNow.Hour >> 1)
);
byte date3 = (byte)((dtNow.Hour << 7) | (dtNow.Day << 2) |
(dtNow.Month >> 2));
byte date4 = (byte)((dtNow.Month << 6) | (dtNow.Year -
2000));
return new byte[] { date1, date2, date3, date4 };
}

```

Výpis 4: Funkce pro nastavení data a času ve čtyřech bajtech

Po naimplementování těchto dvou vrstev komunikačního protokolu RS485, jsem si vytvořil consolovou aplikaci, pomocí které jsem postupně zkoušel zasílat výzvy na získání dat nebo nastavení data a času na měřicí stanici. Po zaslání výzvy a zpracování jí Inmatem, přijde odpověď, která se zpracuje pomocí události Datareceived. Přijaté data jsem následně zpracoval podle typu, zda šlo o čas a datum, systémové proměnné, výpočtové proměnné a nebo o sumy.

4.3 Vytvoření vlastní ORM pro ukládání a získávání dat z firemní databáze

Mým dalším úkolem, který jsem dostal během svého působení na praxi, bylo seznámení se z firemní databází, a vytvoření ORM, pomocí které bych mohl přistupovat a upravovat data o měřících zařízeních typu Inmat. Mezi mnoha tabulkami, které byly ve firemní databázi, mě zajímalo pouze pět tabulek, které jsem využíval. Byly to tabulky Device, Location, Profile, Measurement_Actual a Measurement_History. Jelikož by bylo pro mé účely zbytečné si kopírovat celou databázi firmy na svůj počítač, tak jsem si navrhnul a vytvořil vlastní databázi. Návrh tabulek jsem provedl pomocí nástroje Oracle SQL Developer Data Modeler. Navrženou databázi jsem poté vytvořil a naplnil potřebnými daty z firemní databáze pomocí programu Microsoft SQL Server Management Studio.

ORM je programovací technika, která zpřístupňuje relační data pro objektové prostředí např. C#. Mezi jeho výhody patří, že je velmi dobře přenositelný mezi různými SŘBD, umožňuje práci s objektovým modelem a je rychlejší pro vytváření aplikací. V mém případě jsem si zvolil třívrstvou architekturu ORM [9]. Mezi tyto tři vrstvy patří:

- Prezentační vrstva - jedná se o část, která je viditelná pro uživatele, zajišťuje vstup požadavků a prezentaci výsledků. Je závislá na platformě. V mém případě jako prezentační vrstva je výsledná služba.
- Aplikační vrstva - vrstva, která zajišťuje výpočty a operace prováděné mezi vstupně-výstupními požadavky a daty. Bývá nazývána také Business logika.

- Datová vrstva - jedná se o nejnižší vrstvu třívrstvého modelu, zajišťuje práci s daty. Poskytuje systém řízení báze dat a základní datové operace, které provádějí ukládání, upravování, mazání a výběr požadovaných dat.

Následně jsem tedy vytvořil ORM. Začal jsem tvorbou druhé vrstvy, tedy business logiky. V této vrstvě je složka DataObjects, ve které jsem vytvořil objekty potřebné pro všech pět tabulek. Tyto objekty jsem nazval stejně jako tabulky v databázi. Ke každému objektu je v této vrstvě třída, která volá funkce pro výpis, aktualizaci nebo přidání záznamu z datové vrstvy. Následně jsem naimplementoval třídní knihovnu pro práci s daty v třívrstvěm modelu. V této vrstvě se nachází třída pro samotné připojení k databázi a třídy, které mapují přístup k datům pro jednotlivé tabulky databáze. Po vytvoření těchto dvou vrstev jsem otestoval pomocí consolové aplikace její funkčnost a všechny nedostatky jsem doladil.

Funkce této konečné architektury probíhá takto:

1. Z prezentační vrstvy, tedy ze služby se pošle příkaz na výpis, aktualizaci nebo přidání záznamu.
2. Tento příkaz jde na business logiku, která zavolá příslušnou funkci pro získání dat v datové vrstvě.
3. V datové vrstvě se provede připojení k databázi a provede se požadovaný příkaz nad databází.
4. Z datové vrstvy se vrátí požadované data do business logiky, v případě výběru dat se vrátí DataTable.
5. V business logice dojde k rozparsování dat v podobě DataTable na jednotlivé řádky, které se uloží do listu pro daný objekt.
6. List hodnot pro požadovaný objekt je k dispozici ve službě pro další zpracování.

4.4 Vytvoření výsledné služby pro obsluhu a sběr dat na zařízeních Inmat 51 a Inmat 66

Mým posledním úkolem na praxi bylo tedy vytvoření výsledné Windows služby pro sběr dat na úrovni bajtové komunikace pro měřicí zařízení Inmat 51 a Inmat 66. V této službě jsou využity všechny předešlé úkoly, doplněné o další potřebné funkce.

Popis funkce: Po spuštění služby se spustí nové vlákno, které načte nejdříve z databáze seznam měřících zařízení a jejich profilů pro měření dat do listu, podle lokace a typu zařízení, které jsou nastavené v konfiguračním souboru. Načítání těchto dat probíhá pomocí ORM, které jsem vytvořil v předešlém úkolu. Následně se s listu měřících zařízení vyselektují pomocí jednoduchého LINQ příkazu všechny COMporty, přes které se připojují k počítači. Pro každý takhle získaný COMport, se vytvoří nové vlákno, do kterého se načtou s listu měřících zařízení všechny

potřebné informace pro otevření sériového kanálu pro komunikaci pomocí RS485. Když je komunikační kanál otevřen, přejde služba k zasílání žádostí na jednotlivé měřicí zařízení postupně pro každý jednotlivý COMport. Dále dojde k vyvolání funkce, která má každou hodinu zkontrolovat aktuální čas a porovnat ho z časem na měřicím zařízení. Jsou-li tyto časy rozdílné, tak dojde k vytvoření příkazu, s nastavením aktuálního data a času a poslání ho pomocí komunikačního kanálu na daný měřák. Po této kontrole už následuje zasílání jednotlivých výzev pro získání dat z měřáku, a to tak, že se pomocí LINQ příkazu získají všechny měřicí profily pro aktuální zařízení. Z těchto profilů se načte fáze, podle které se zašlou potřebné příkazy pro získání systémových proměnných, výpočtových proměnných nebo sum. Po zaslání každé výzvy následuje přečtení odpovědi na tuto výzvu, která přišla po sériovém kanálu zpět. Čtení odpovědi se provádí opět podle fáze, tak aby bylo možné přečíst jednotlivé data. Následně probíhá zápis těchto dat do databáze. Tyto údaje o měřeních se zapisují do tabulky Measurement_Actual, kde dochází k přepisování aktuálních dat při každém měření, nebo do tabulky Measurement_history. Do této tabulky se zapisují data vždy každou čtvrt hodinu. Ještě než se data zapíší do některých s těchto dvou tabulek, však dochází ke kontrole, zda se mají data pouze zapsat, nebo se s nich mají vypočítat průměrné či maximální hodnoty. Tato kontrola proběhne podle stavu, který se opět získá pomocí LINQ příkazu, tak že se získá pro daný měřák s jeho aktuálního profilu stav, v kterém mají být data uložena. Po úspěšném uložení aktuálně získaných dat se pokračuje následujícím zasláním výzvy, nebo dojde k přepnutí na další vlákno v pořadí s následujícím COMportem.

Všechny tyto operace, prováděné službou jsou odpovídajícím způsobem ošetřena a zapsána do logovacího souboru. Pro logování jsem použil Nlog, který je možné jednoduše nainstalovat do Visual Studia. V Nlogu jsem si nastavil pomocí konfiguračního souboru výstup do console, která se používá při testování funkčnosti, a dále do čtyřech souborů, kde zapisuje samotná služba. Tyto soubory byly určeny pro debugovací účely a informační, varovné a chybové výpisy. Díky tomuto rozdělení je o dost jednodušší následné hledání chyby při budoucím použití v reálném provozu.

5 Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe

V průběhu mého působení na praxi během plnění těchto všech úkolů jsem využil poměrně dost znalostí nabytých v průběhu studia. Mezi hlavní znalosti, které jsem využil, bylo programování v C#, jenž jsem se naučil díky předmětům Programovací jazyky II a Architektura technologie .NET. V tomto předmětu jsem se naučil základní technologie na platformě .NET jako je práce s vlákny, základy LINQu a zejména vytváření Windows služby, což byl můj hlavní úkol na této praxi. Dále jsem použil poznatky při tvorbě ORM z předmětu Databázové a informační systémy a zejména třívrstvou architekturu, kterou jsme používali ve Vývoji informačních systémů. Pro SQL příkazy použité pro získání dat v ORM jsem vycházel s poznatků nabytých v předmětu Úvod do databázových systémů.

6 Znalosti či dovednosti scházející v průběhu odborné praxe

V průběhu vykonávání této odborné praxe mi scházely zejména znalosti ohledně měřících zařízení Inmat 51 a Inmat 66. Tyto informace jsem musel nejdříve nastudovat z různých manuálů, a to zejména informace o komunikaci po komunikační lince RS485 s těmito měřícími zařízeními. Dále jsem z počátku nevěděl, jak má vypadat služba, jenž se stará o sběr dat v reálném provozu, pro více různých zařízení pracujících zároveň. Tyto informace jsem se však později dozvěděl pomocí nastudování této problematiky a hlavně díky rad, které mi poskytnul můj vedoucí ve firmě.

7 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

V této bakalářské práci jsem se snažil popsat co nejpřesněji mou odbornou činnost a plnění úkolu, zadaných během působení na odborné praxi ve firmě Autel, a.s. Povedlo se mi vytvořit Windows službu, která se stará o sběr dat z měřících zařízení Inmat 51 a Inmat 66 a tyto data následně ukládá do firemní databáze. Služba pracuje zcela samostatně na pozadí a zapisuje svoji činnost do logovacího souboru.

S mým působením na praxi jsem velice spokojen a to zejména díky tomu, že jsem mohl nabýt nové zkušenosti s prací na reálných projektech ve firmě, což bude mít pro mě do budoucího zaměstnání jistě velmi pozitivní přínos. Také jsem získal hodně cenných rad od vedoucího bakalářské praxe ve firmě, a to ohledně pracování na firemních projektech až po působení v IT firmě celkově.

Literatura

- [1] Stránky firmy Autel, a.s. [online]. [cit. 2016-08-04]. Dostupné z: <http://www.autel.cz/>
- [2] Katalogový list Inmat 51 [online]. 2009 [cit. 2016-08-04]. Dostupné z: <http://www.zpanp.cz/files/doc/matematicky-clen-pracovni-meridlo-inmat-51-24-4.pdf>
- [3] Návod k výrobku Inmat 51 [online]. 2014 [cit. 2016-08-04]. Dostupné z: <http://www.zpanp.cz/files/doc/matematicky-clen-pracovni-meridlo-inmat-51-41-1.pdf>
- [4] Obrázek Inmat 51 [online]. [cit. 2016-08-04]. Dostupné z: <http://www.zpanp.cz/fotografie/matematicky-clen-pracovni-meridlo-inmat-51-159-0.jpg>
- [5] Katalogový list Inmat 66 [online]. 2009 [cit. 2016-08-04]. Dostupné z: <http://www.zpanp.cz/files/doc/meric-prutoku-tepla-stavovy-prepocitavac-plynu-inmat-66-24-4.pdf>
- [6] Návod k výrobku Inmat 66. ZPA Nová Paka. 2009
- [7] Obrázek Inmat 66 [online]. [cit. 2016-08-04]. Dostupné z: <http://www.apoelmos.cz/images/products/504/full/466.jpg>
- [8] Popis komunikačního protokolu RS485 Inmat 66 a Inmat 51, verze 3.00 [online]. 2009 [cit. 2016-08-04]. Dostupné z: <http://www.zpanp.cz/files/doc/matematicky-clen-pracovni-meridlo-inmat-51-8.pdf>
- [9] Třívrstvá architektura [online]. 2011-2013 [cit. 2016-08-04]. Dostupné z: <https://managementmania.com/cs/trivrstva-architektura-three-tier-architecture>

8 Přílohy

I. Příloha na CD

Příloha obsahuje složku Inmat51_66_service, v které se nachází zdrojový kód výsledné služby.